Requirements for a successful implementation of Oracle Fusion

Session # S300002

July 18, 2008
Prepared by Frank Bommarito
DBA Knowledge, Inc.

Table of Contents

INTRODUCTION

Read this paper if an installation or upgrade of Oracle Fusion products is on the horizon. By reading this paper, the reader will become aware of what is required to have effective and complete installation architecture.

Oracle Fusion is the melding of middleware product offerings from Oracle. These applications include Forms, Reports, Portal, OID, SOA, BPEL, ESB, BAM , and many others. There are many times when more than one of these applications is implemented together. Fusion Applications is one of the growth areas with Oracle and truly allows Oracle to be a leader in the software field. Oracle Fusion applications, more often than not, utilize an Oracle database to store and access data. While the knowledge for setting up an Oracle database to be flexible and robust is wide spread, the same knowledge is not so wide spread for Fusion Applications. Many third-party and Oracle based tools exist for management of databases, but, few third-party tools exist for management of Fusion Applications.

It is common to implement Oracle Fusion without a complete set of operational procedures. Operationally, Fusion products can be maintenanced just as easily as the Oracle Database. This paper will highlight some of these operational areas to ensure that they are not overlooked. This paper will also illustrate the impacts of implementing properly versus just implementing.

In the following paper, there are some technical examples. These examples were physically used and represent the best knowledge at the time of writing this document. Usage of any of these commands is at the sole discretion of the reader. Adjustments must be made to match host machines and protocols required where the deployment is occurring.

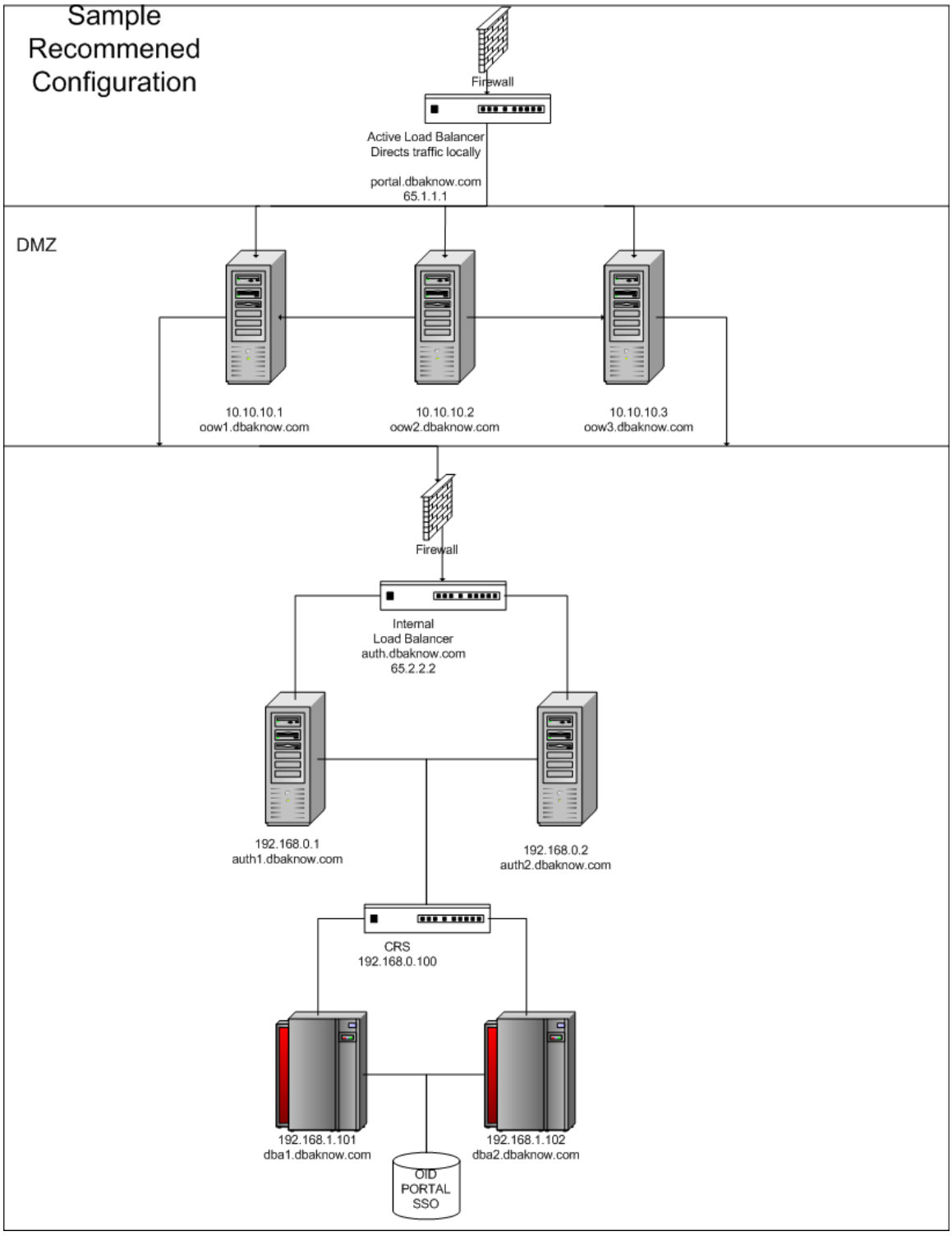
REQUIRED COMPONENTS

Most Fusion applications utilize at least one end user application, authentication application and database. The required components are best shown utilizing an example. In this example, the application will utilize Oracle Portal, Oracle SSO, and Oracle OID with an Oracle RAC database backend. The sample configuration encompasses all components for smaller and/or larger installations.

The following table represents the Oracle Homes that are required:

Product	Needs
Oracle RAC	Database Software Cluster Software ASM or OCFS or equivalent
Oracle OID (LDAP)	Application Server software Database Connection
Oracle SSO	Application Server software Database Connection
Oracle Portal	Application Server software Database Connection

A diagram of the above is shown here:



While some products within the Fusion stack can share executables, it is typically recommended to share in only a few cases. Sharing reduces installation time; however, sharing will delay patching and flexibility down the road. Databases can also be shared. Specific recommendations on sharing database are much more a question of how the application will be deployed as opposed to a specific best practice.

Note that Oracle has multiple versions of the application server software. Each version supports different tools. In the example above (at the time of this writing), 10.1.2.0.2 is required (as a base level before patching) for the product shown.

Outside of the Oracle components, the following is also required:
Host machines
Application Context Switch (or equivalent software)
Firewall (Assuming any outside presence of an application
Appropriate disk space for installation
Dedicated network switch for RAC
Tape subsystem (or other backup device)

How many of each of these components is dependent upon the specific deployment. Popular choices are discussed later in the paper.

WHO MUST BE INVOLVED?

The components that are required – require participation from a wide variety of IT departments. These departments may or may not be physically near each other. These departments may or may not have direct contact with each other. These "may or may not's" can cause troubleshooting problems down the road and should be addressed. If each specific department looks only at the area where they are normally charged, then, finding a problem could prove to be difficult. This stems from the fact that no single person or group will have knowledge of the flow of data. Often problems appear larger than they are because no single group can find any logical issue. There in lies what is known as a "Fusion DBA". For those familiar with Oracle Applications and the concept of an "Applications DBA", this term (Fusion DBA) should be easy to understand. A Fusion DBA is someone who has an Oracle based background that understands the Oracle Fusion architectural components. The Fusion DBA will be an integral part of the design, setup, implementation and operational maintenance of a Fusion deployment.

If a deployment is broken down from "100%" of the total time required then, the table below represents the IT positions required and the percentage of time needed for the project duration during design and deployment:

IT Position	Percent of project	Typical Duties
Network Administrator	10	Setup application context realms. Create virtual IP addresses Setup specific routes and network rules Create and publish DNS entries Adjust virtual IP death detection tests
System Administrator	15	Build host machines (could use GRID) Allocate disk Execute root commands as required Setup backup systems
Fusion DBA	50	Coordinate with all other resources Install Oracle software Create GRID beacon tests Ensure operational checklists are complete
Developers	20	Deploy application or call web services Note – time to develop the application is outside of the deployment,
Cross team management	5	Ensure that needs of other teams are prioritzed as every single wait has large

		hours of down stream impact. An example is when a DBA must have a ROOT script executed; waiting here could delay every other phase of the project – could push meetings and run into schedule off time. As well as such waits typically lead to budget overruns.

Senario One

At one time DBA Knowledge was called upon to execute on the following:
Install Portal
Install Portlets for JDE Enterprise One
Synchronize OID from AD
Advice on best practices
Ensure that items were migratable from Dev through Proudction.

During this consulting assignment, a request was made for the team members above. The AD administrator was unable to attend. It was pointed out during the discovery time that the lack of participation could cause problems with AD syncing. The requirements were specificed during this meeting.

While configuring the synchronization routimes, the AD person objected to the previously stated requirements. While the objections were valid, the timing of the objections caused some siginfiicant scheduling issues. The project was still completed on-time; however, some training time had to be sacrificed to meet the deadlines.

The moral here – even the smallest items must be represented at the beginning.

COMPLETE THESE – BEFORE - STARTING

There are several activities that, if not done ahead of time, will likely require rework. Any rework requirements will have a multiplying impact on the total estimated time for the project. With Fusion applications, many installation steps have dependencies on prior installation steps. If on the fifth step, it is realized that step two was not completed properly, step two, three, four must be completed again before step five can be executed. This can cost many hours or days. The items that have this capability are at the base of any installation. While most of these items can be changed post-deployment, they are difficult to change while deploying. With a standard database, if an installation starts and a default character set is used – after some time it may be discovered that a different character set was required. This kind of change often causes significant time and dollars. With Fusion, the same dependencies exist. The following is a list of common – must have – pre-installation requirements:

Component	Description
DNS names	What URL will be used to access the applications (i.e. Portal, SSO and OID, BPEL)
Port Numbers	What port numbers must be opened? Without knowing this, it is often found that post-installation the network team will not allow what the Fusion DBA implemented. Some things are easier to change then others. (i.e. 80 – 443)
IP Address of the host machines	Do not utilize temporary domains and addresses. It is often easier to re-install than it is to change IP address with the

	current software available. If using a Load Balance Router(LBR), changing names is much simplier.
What software to install	To be able to publish all needed URLs and Ports numbers, the Fusion DBA must be aware of what is to be installed. Adding other applications often requires more ports – more disk – more function setups – etc… Not all Fusion products are available in the same release of the software.
Installation Standards	Host names, packages, directories, password schemas, Linux user and group IDs (where required), location of scripts and others. If the organization does not have standards, then, create them. Some standards are shown in this paper for suggestive purposes alone. Consistency is the key!
How many?	How many machines will be used in the deployment? Will VMWare be used? Will SSO and OID be installed together or separate? If items are chosen to be installed together, is it envisioned that these items will ever be split?

HOST NAMES

All applications will have a least two host names. One host name will be to determine physical attributes (the actual machine) and one host will be for virtual concepts (the LBR). The application will be accessed from the LBR and not directly from the machine. Implementing as such will allow for great flexibility as the system expands. A LBR should be used and its usage should contain a separate URL for each Oracle installed home product (i.e. portal.dbaknow.com – bpel.dbaknow.com) even if those products are installed on the same physical host machine.

This is the seven total host machines. The following table lists a sample of host name:

LBR VIP	LBR Front End	Hostname	Machine Purpose
LBR	Portal.dbaknow.com	oow1.dbaknow.com	Portal server
LBR	Portal.dbaknow.com	oow2.dbaknow.com	Portal server
LBR	Portal.dbaknow.com	oow3.dbaknow.com	Portal server
LBR	Auth.dbaknow.com	Auth1.dbaknow.com	OID/SSO Server
LBR	Auth.dbaknow.com	Auth2.dbaknow.com	OID/SSO Server
VIP	Dba1.dbaknow.com	dboow1.dbaknow.com	RAC Server 1
VIP	Dba2.dbaknow.com	dboow2.dbaknow.com	RAC Server 2
N/A	LBR Not Needed	Dev.dbaknow.com	Development

Senario Two

At one time DBA Knowledge was called upon to execute on the following:
Install Forms
Install OID
Advice on best practices
Migrate older version of Oracle forms (6i) to the new version of Oracle forms (10g)

All appropriate parties and items were identified ahead of time. Impact analysis was completed on the applications. The development (single node) server was completed and tested without problems. While production was being installed, the installation would not complete. The stalling occurred while OID was attempting to configure itself with the LBR. The required networking staff was not available for assistance in a timely manner. The problem was determined to be that the LBR was preconfigured to be "off" until the software was installed. The software would not install because of the LBR status. After one week, the network personnel disabled the health checks allowing the project to resume.

The moral here – the project team – all parties – must be involved from the beginning to the end (or at least accessiable).

INSTALLATION

Installation must be done at the lowest levels and worked up the stack. After each installation, ensure to take a backup prior to the next major component. Also, test all base functionality before proceeding.

The order of installation depends on the products that are being installed. After completing an application architecture overview, the order of installation should be clear. In the pictured application (as diagramed above), the Oracle RAC database is the first Oracle component; however, the physical host machines, LBR, firewall and related settings must be completed first.

Prior to starting the installation of Oracle products, ensure that required components are in place. Take some time to validate the installation against Oracle requirements.

Oracle RAC – Switch, dual networks, bonding, redundant storage paths – etc…
Oracle Application Servers – LBR, firewall rules, SSL accelators, etc…

Oracle product installation is completed in the following order:
Oracle CRS
Oracle ASM
Oracle RAC
Oracle MRCA
MRCA Creation
Oracle OID
Oracle SSO
Oracle Portal

Many of the Fusion products will install or utilize a database created by the Fusion product; however, in this installation the DBA team will create a separate database specifically to house the Portal, SSO and OID repositories. This is completed with the MRCA (Meta Repository Creation Assistant). A separate database is chosen so that the database can follow existing company standards as opposed to the default database which is not likely to be of a standard install.

Senario Three

In an effort to save time on an installation, the Fusion DBA assumed that the installation would go without issue. The DBA installed RAC without any issues. The MRCA created without any problems. OID/SSO was successfully installed. When Portal was installing, a few problems occurred with the network while the installation was in progress. These issues caused the Fusion

DBA to abandon the installation. Oracle Portal had "partially" installed in the database. There was a several day process to determine the best course of corrective action. SR's were created with Oracle. In the end, the database had to be recreated and the application server produts had to be reinstalled. The installation process takes about 8 hours for all items discussed (with backups). The same installation takes about 6 hours without backups. This install took 1 week.

The moral here – never abandon operational principles for the sake of time. Doing so will cost time in the long run.

Machine Preparation

Wheere installing Oracle RAC or any application server product, a series of setups must be completed on the host machines. These setups are required for Physical and Virtual machines. Even when the requirements seem obvious and/or not "really" required, complete the steps.

As an example – for Oracle RAC – the following steps must be completed prior to beginning the installation:
Identify Host Names/IP Address
Ensure that the servers have their time synced
Identify the storage devices that will be used
Identify if RAW, OCFS2 or NFS will be used for Quarom disks
Ensure that the disks are read/writable across nodes
Ensure that user id and group ids are consistent across nodes
Ensure that users can communicate with each other without a password
Ensure that Oracle OS settings and requirements are complete – and – persistent on reboot

After all of this is in effect, that is the time to begin the installation.

Senario Four

At a customer site, an experienced DBA was installing a "solid" production environment. The DBA started the installation process and "assumed" that system setups were correct. A job that was bid out to complete in 8 hours took almost 40 hours. Each step along the way ran into issues. From disk writing errors to network interface errors. The customer left with the false impression that Oracle RAC was not ready for "prime time".

DBA Knowledge came into the environment to install a "new" RAC cluster and migrate the installed set. This was done because of a series of unexpected reboots that were occurring. A list of pre-install steps were completed with the system administrator. From when/how to use /dev/mapper to the purpose of the Virtual IP addresses. Once complete, the installation took very little time (about 3 hours total). This installation included the "known" patch lists. This customer has been 100% uptime since the installation and their perception of Oracle has changed to the positive.

The moral here – taking a few extra moments to properly prepare will save time, frustration, blood pressure and perfection. If it's worth doing, it's worth doing right.

Oracle RAC

Oracle RAC is installed with three different network addresses.
Public is the IP address (with associated DNS entry) that is commonly used to access the machine.

Private is the IP address (typically not in DNS – non-routable) that is used for the cluster interconnect (1.1.1.1 is not uncommon for such an address).
Virtual is the IP address (with associated DNS entry) that is used to connect to the database. This name will be used in any OCI or JDBC connection string. Oracle manages this IP address meaning that it cannot be started by the operating system at boot time.

In our case – the following table represents the Names+Address that are used:

Address Type	Address
Private	1.1.1.1 (No DNS) 1.1.1.2 (No DNS)
Public	dboow1.dbaknow.com dboow2.dbaknow.com
Virtual	Db1.dbaknow.com Db2.dbaknow.com

Ensure that the "Virtual" names are generic and used in a similar fashion as the LBR names are used. A common mistake is to name the virtual names after the host machine. If the host machine is dk01pm.dbaknow.com – a common virtual name would be dk01pm-vip.dbaknow.com. The problem with using this type of name is that "dk01pm" has special (smart key) meaning. If any of the smart key assumptions are ever changed, then, much re-work may be required. A common change would be a move from one data center to another (dk01pm – Park Meadows -- becomes dk01nl – New Location). If this change occurs, then, connection strings JDBC, OCI, ODBC must all be modified. Some of these connections strings may not be easy to change. Using a virtual name such as "db1.dbaknow.com" removes this problem. If a new location is needed, simpy restore the databse into the new location, modify the DNS setting for db1.dbaknow.com and the transistion is complete. The virtual name for RAC is similar to the LBR name for the application server products.

After installing RAC, perform cluster based failover testing, network card failure testing, and switch failure testing and the like – before proceeding. Once complete, execute a full backup before moving on.

Senario Five

An Oracle RAC installation was complete. Failover testing was also complete. The application was moved into production. A few days later, an machine in the cluster crashed and the crash was forced by Oracle CRS. The application survived because of the failover testing; however, management wanted to discuss the problem. The actual crash was caused by a known software bug. This bug also had a known fix. If a few moments had been spent, during the installation process, to determine and check for known issues, then, a patch would have been installed before going live. In this case, one week after go-live, the cluster had to be taken down to apply a series of patches. The new highly redundant system had exceeded its yearly downtime in one week.

The moral here – downloading, installing, and testing the base software is not sufficient. Always take some time to check for known problems.

MRCA

MRCA (Metadata Repository Creation Assitant) will require a pre-created (typically empty) database. MRCA will create the schemas required for Oracle Fusion products. Ensure to download and install using the appropriate MRCA software (correct release). It is important to

note that the MRCA will require that the Ultra Search option is installed in the database. If a database was created without the Ultra search option, then, utilize "dbca" to add this option before installing the MRCA software (A general purpose DBCA created database will not include the ultra search option by default).

After installing the MRCA software (either through the installer or post-install), run the Repca assistant:

```
$ $ORACLE_HOME/runRepca

Have handy SQL Net connection information and the Oracle Home name for the RAC database.
The installer will prompt for an action – choose "Load" when prompted.
```

After this point, ensure that a database backup is taken so that a rollback is easy should it be needed (RMAN or Restore Point is recommended).

```
$ sqlplus "/as sysdba"
SQL> create restore point post_mrca;
```

A restore point is a 10g Oracle database feature that requires some configuration prior to implementing. A restore point also requires that a database backup be taken at some time prior to the restore point. The reader must ensure that all requirements are met prior to relying on this functionality. An alternative is a whole database backup.

CONFIGURATION

Other Oracle Products

The other Fusion products will require (at some point) a connection string to connect to the database. The other Fusion products will have minimum Oracle initilzation parameters that are in effect prior to installation. To avoid delays, ensure that Oracle initialization parameters are in effect prior to installation. Gather required connection string information prior to installation. Recall that if a RAC environment is used, then all connection string information should account for failover strategies.

If the intent is to install in a clustered manner (i.e. many Oracle Portal hosts), then, the Network administrator must be involved during the installation. A LBR contains an item called a "Health Check". A single LBR points to more than one IP address. One of the IP addresses is disregared if a service fails a "health check". If all health checks are disabled, then, both IP addresses will respond during the installation. If the health checks are enabled, then, neither IP address will respond during the installation as the required services are not yet installed. Before installing the first node, have the network administrator disable one of the two IP address. When installing the second node, ensure that both IP addresses are enabled. Once the install is complete – tested – and backed up, have the network administrator enable all health checks.

Below are examples of the updates TNS and SQLNet.ora files – these entries account for LDAP (OID) and failover settings.

```
# A TNS Name that handles base failover for RAC
DB,DB.dbaknow.com =
  (DESCRIPTION =
    (LOAD_BALANCE=ON)
    (FAILOVER=ON)
    (address = (protocol = tcp)(host = db1.dbaknow.com)(port = 1526))
```

```
    (address = (protocol = tcp)(host = db2.dbaknow.com)(port = 1526))
    (CONNECT_DATA =
      (SERVICE_NAME = db.dbaknow.com)
    (failover_mode =
      (backup = db.dbaknow.com)
      (type = select)
      (method = basic)
      (retries = 20)
      (delay = 5)
      )
    )
  )
```

sqlnet.ora:

```
NAMES.DIRECTORY_PATH= (TNSNAMES, LDAP, EZCONNECT, ONAMES, HOSTNAME)
USE_DEDICATED_SERVER=ON

NAMES.DEFAULT_DOMAIN = dbaknow.com
```

For JDBC thin client connections, the following example should be used:

```
jdbc:oracle:thin:@(description=(load_balance=yes)(failover=yes)
(address=(host=db1.dbaknow.com)(protocol=tcp)(connect_data=(service_name=db.dbaknow.com)))
(address=(host=db2.dbaknow.com)(protocol=tcp)(connect_data=(service_name=db..com))))
```

For all Products (except for OID) – utilize the following example ldap.ora file:

```
DIRECTORY_SERVERS= (auth.dbaknow.com::636)

DEFAULT_ADMIN_CONTEXT = "DC=TNS,DC=DBAKNOW,DC=COM"

DIRECTORY_SERVER_TYPE = OID
```

After installing each and every component – utilize the ROOT user and backup the installed execuables (bring down the product first):
For all Products (except for OID) – utilize the following example ldap.ora file:

```
$ $ORACLE_HOME/opmn/bin/opmnctl stopall
$ su – root
$ cd $ORACLE_HOME
$ cd ..
$ tar cf /save_install.tar *
```

Senario Six

During an installation, the network admistrator stated that it was not possible to adjust LBR settings. The LBR health checks were disabled, the first OID install worked; however, when the second node was attempting to join the cluster – that second node received errors. These errors were occurring because the first node was responding when the second node needed to respond.

DBA Knowledge spoke with the network administrator and suggested that modifying the LBR settings to otherwise available and not-avaialble ports would accomplish the need. The network administrator made the changes and the installation resumed – and – completed without further problems.

The moral here – any statement that has “its not possible” in it should be questioned.

Architecture

It is typical to have an installation that has redundancy for Production and no redundancy for development. The “middle” environments (i.e. test) are typically up for debate and are dependent upon budgets.

When a production environment is proposed with redundancy, this should be completed at every level. The reasoning here is that the purpose of redundancy is uptime. If an installation promises great uptime, then, that installation cannot be dependent on other systems that do not promise the same level of uptime.

Senario Seven

An Oracle OID for Enterprise User Security project was beginning. During the architecture phase, the client insisted on redundant components for every level. Ironically, this was carried through except for the front-end – the client did not want to utilize a LBR. A DNS router was the choice (a single DNS lookup that returns one of many addresses without any health checks). This would have increased the chances of downtime and could cause “hangs” in crucial applications that would be dependent on the system. Fortunately, the client listened to reason and prevented this fatal mistake.

The moral here – doing half an installation is not sufficient. Commit to go all the way or do not start.

ADMINISTRATION

LBR Health Check Monitoring

There are three main load balancing router (LBR) configurations that are utilized in the diagramed example. There is one for the Portal midtier server, one for the OID server and one for the SSO server. Even through the SSO and OID server are installed on the same hosts, they listen on different ports and can have different rules. Each LBR runs through a series of health check monitoring. These checks allow for servers to automatically be added to and taken from an LBR farm. The current configuration limits each server that is installed to two health checks. LBR farms contain at least one and up to many servers within the farm.

The health checks are all done concurrently (meaning there is no specific order). A common practice for health check frequency is about one minute. Any faster can cause performance issues and any slower can lead to long delays in detecting a failure.

The following tables show what is to be monitored:

Portal Servers (LBR=portal.dbaknow.com)

http://oow1.dbaknow.com:7787
http://oow1.dbaknow.com:7788
http://oow2.dbaknow.com:7787
http://oow2.dbaknow.com:7788

```
http://oow3.dbaknow.com:7787
http://oow3.dbaknow.com:7788
```

SSO Servers (LBR=auth.dbaknow.com)

```
http://auth1.dbaknow.com:7777/pls/orasso/orasso.home
http://auth1.dbaknow.com:7777/oiddas
http://auth2.dbaknow.com:7777/pls/orasso/orasso.home
http://auth2.dbakbnow.com:7777/oiddas
```

OID Servers (LBR=auth.dbaknow.com)

```
ldap://auth1.dbaknow.com:389
ldap://auth2.dbaknow.com:389
```

TNS Entries into LDAP

The primary source for TNS Names resolution is LDAP (OID). The following steps will load all necessary entries into the LDAP server.

From any operating authorization server (OID Server), setup the environment and ensure that X-Windows can operate. This command set reads an existing TNS Names.ora file and loads them into LDAP:

```
$ cd $ORACLE_HOME/network/admin
$ netmgr (GUI Tool)
        Command
        Directory
        Export Net Service Names
        Enter Credentials
Next
Doman=dbaknow.com
        Choose all TNS Names
        Next
        Next
        Finish
        File
        Exit
```

LDAP Entries into TNS

There are no currently provided utilities from Oracle that dump TNS entries stored in OID to a tnsnames.ora file. The shell script here does a pretty good job of filling the gap.

```
ldapsearch -p 389  -h auth.dbaknow.com -D cn=orcladmin -w Manager1 -b
"cn=OracleContext,dc=dbaknow,dc=com" -s sub objectclass=*  cn orclnetdescstring  |grep -v
",cn=OracleContext," > OID.dump


oIFS=$IFS
IFS=${IFS#??} #Just a newline, nothing else.
DL_WORK=0
```

```
for DL_LINE in `cat OID.dump`
do
  DL_IS_IT=`echo ${DL_LINE} | grep -v "^orclnetdescstring=" | wc -l`

  if [ ${DL_IS_IT} -gt 0 ] # It is not an orclnetstring
  then
    DL_IS_CN=`echo ${DL_LINE} | grep -v "^cn=" | wc -l`
    if [ ${DL_IS_CN} -gt 0 ] # It is not a CN
    then
      DL_WORK=0
    else
      DL_WORK=1
      DL_LINE_CN=${DL_LINE}
    fi
  else # It is a netstring and we have a CN
    if [ ${DL_WORK} -eq 1 ]
    then
      echo ${DL_LINE_CN} | awk -F= {'print $2" = "'}
      echo ${DL_LINE} | sed 's/^orclnetdescstring=/+/' | awk -F+ {'print $2'}
      DL_WORK=0
    fi
  fi
done

IFS=$oIFS
```

Move Content from Dev to Prod or elsewhere

It is important to understand how to migrate data from system to system. The examples shown here move all data (from different tiers) into development. Each individual area is separated and can be moved independent of the other.

Most of the moving is complete via a series of small shell scripts (referenced below and available in Apendix A for this document). As with any coding example, some modifications may be necessary for actual usage.

All examples here are for the Portal product. Each Fusion product has separate instructions for migration. Refer to the documentation of each product.

PORTAL

The following table should be filled out prior to beginning.

Item Needed	Example	Description
Target Host Name	Development.dbaknow.com	The portal host machine
Target Port Number	7777	The webcache listen port
Source Host Name	Portal.dbaknow.com	The portal host machine (or LBR if used)
Source Port Number	443	The webcache listen port
Target Oracle Home	/oracle/product/as_ptl/10.1.4	The install path to portal
Target Database SYS Password	Manager1	Into the portal database
Target TNS Name	repdev	Into the portal database

Target – Passwords for: PORTAL PORTAL_APP PORTAL_DEMO PORTAL_PUBLIC		Can be retrived via this script: get_password.sh

Populate a file called "/oracle/migrate/portal/portal.env" with the above information. File should look similar to this:

```
unset NLS_LANG
export DEV_HOSTNAME=development.dbaknow.com
export DEV_PORT=443
export DEV_DN="dc=dbaknow,dc=com"

export PROD_HOSTNAME=portal.dbaknow.com
export PROD_PORT=7777
export PROD_DN="dc=dbaknow,dc=com"

export MIDTIER_ORACLE_HOME=/oracle/product/as_ptl/10.1.4

export SYS_PASSWORD=manager1
export PORTAL_TNS=repdev
export PORTAL_PASSWORD=LdxPs709
export PORTAL_APP_PASSWORD=CyuO4onm
export PORTAL_DEMO_PASSWORD=H0L8Kuwg
export PORTAL_PUBLIC_PASSWORD=PdGD6yeX
export PORTAL_USER=portal
```

After creating the "/oracle/migrate/portal/portal.env" file, ensure that this file has execute permission:

```
$ chmod 750 /oracle/migrate/portal/portal.env
```

PRE-STEP TO BE COMPLETED BY DBA TEAM ON SOURCE DATABASE (REP) SERVER:

```
# Utilize a DBA account (i.e "/ as sysdba")
$ exp file=portal_exp.dmp grants=y log=portal_exp.log owner=portal,portal_app,portal_demo,portal_public
```

Once this step is complete, have the DBA place the file "portal_exp.dmp" file in a location that it can be retrieved.

Copy the file to the target database server:

```
$ ssh development.dbaknow.com (as oracle)
$ scp db1.dbaknow.com:portal_exp.dmp /oracle/migrate/portal/exp_data
```

Items below are shown in steps. This is done because many steps can be rerun in case of errors and each step should be successfully completed before moving on to the next step. Each script has the word "Portal" or "DB" in it. If "DB", then, the ORACLE_HOME must point to the database software. If "Portal" then, the ORACLE_HOME must point to the Portal application srever executables.

STEP ONE
This first step will physically delete all existing portal users from the current (target) environment. Prior to executing, ensure that Portal is NOT running and that no database connetions to the users “portal, portal_demo, portal_app, and portal_public” exist.

Setup the database environment for the Portal installation and then run the script as shown here:

```
$ . oraenv
ORACLE_SID = [as_inf] ? as_ptl
$ cd /oracle/migrate/portal
$ ./First_Portal.sh
```

STEP TWO
This step imports that data from the pre-step into the target database. This can take upto forty minutes to complete.

Setup the database environment for the Database installation and then run the script as shown here:

```
$ . oraenv
ORACLE_SID = [as_ptl] ? infd
$ cd /oracle/migrate/portal
$ ./Second_DB.sh
```

STEP THREE
This step recompiles invalid database objects that can be compiled.

Setup the database environment for the Database installation and then run the script as shown here:

```
$ . oraenv
ORACLE_SID = [infd] ? infd
$ cd /oracle/migrate/portal
$ ./Third_DB.sh
```

STEP FOUR
This step updates the imported data to change host and port names from the source environment to the target environment (i.e. from Stage to Dev)

Setup the database environment for the Portal installation and then run the script as shown here:

```
$ . oraenv
ORACLE_SID = [infd] ? as_ptl
$ cd /oracle/migrate/portal
$ ./Fourth_Portal.sh
```

STEP FIVE
This step combines existing OID data with the newly imported Portal data.

Setup the database environment for the Portal installation and then run the script as shown here:

```
$ . oraenv
ORACLE_SID = [as_ptl] ? as_ptl
$ cd /oracle/migrate/portal
$ ./Fifth_Portal.sh
```

STEP SIX

This step is optional. This step will enable flashback mode on the database. Run this only if that is desired.

Setup the database environment for the Database installation and then run the script as shown here:

```
$ . oraenv
ORACLE_SID = [as_ptl] ? infd
$ cd /oracle/migrate/portal
$ ./Sixth_DB.sh
```

VALIDATION

Restart all tiers and validate access into the application:
http://development.dbaknow.com:7777/pls/portal

Senario Eight

The question of virtualization arose during architecture discussions on a client site. To use or not use – supported or not supported – standard or non-standard. Factually, Oracle supports virtulization using Oracle VM Ware. Oracle's support stance outside of using Oracle virtualization is that a problem would need to be reproduced outside of the virtual environment before calling Oracle. This led the client down a path that would have had them purchase many host machines and abandor utilizing virtualization. If this plan had been followed, the total cost of the project would have been too high and never would have started.

In reality, the defacto industry standard is to utilize virtualization (supported or not) in non-Production environments and to install Production without virtualization. While there is still debate in these areas or on these points, utilizing virtualization can/does/will save time and money (and is greener as there are fewer computers to plug-in – score one for the Earth).

The moral here – research does not end with the first negative statement.

BACKUP/RECOVERY

Backups of application server tiers occur in two major sections – Full and Incremental. Full is done only after major modification to the components such as an upgrade or install. Full backups should be done while the system is not running. Incremental backups are done nightly and can be done while the system is running.

Application Servers

FULL BACKUPS

A full backup should be done with the entire set of executables on the host machine being backed up – down. If the executables are running, then, a restore requires advanced troubleshooting skills in many cases.

The following is necessary to complete a full backup:

```
# It is also recommended to backup the executables – do this as "root".
# This will capture sticky bits and other options that may be missed.
$ su -
```

```
$ cd /oracle/product  # Assumes that this is the ORACLE_BASE
# For each Oracle Home
$ tar cf as_portal.tar as_portal
$ gzip as_portal.tar
```

CONFIGURE BACKUPS FOR APPLICATION SERVERS

Backups for Portal and SSO/OID and all application server tiers must go through a quick configuration step. This is typically completed one time per set of application server executables. The results from these setup steps will be that incremental backups and restores will be possible.

```
$ cd $ORACLE_HOME/backup_restore
$ vi config/config.inp
        Set values for
                log_path=/oracle/admin/as_portal/backup/log
                config_backup_path=/oracle/admin/as_portal/backup/config
# Note that the actual directory names are per local standards.

$ ./bkp_restore.sh -m configure
```

On each node:

```
Command to execute a backup
$ $ORACLE_HOME/backup_restore/bkp_restore.sh -v -m backup_config

Command to execute a restore
$ $ORACLE_HOME/backup_restore/bkp_restore.sh -m restore_config –t 2006-05-11_15-50-07
Where the value for "-t" comes from:
$ ls /oracle/admin/backup/as_portal/config/config_bkp_*
/oracle/admin/backup/as_portal/config/config_bkp_2006-05-11_15-50-07.jar
```

When scheduling backups for the application servers, the backup start times on each node should be slightly staggered (15 minutes). This is because the DCM archive name is based on a timestamp and must be unique across the farm.

Database Servers

Database backups can be accomplished via RMAN or Hot Backups or any other method that is currently employed as a standard.

```
$ rman target /
RMAN> backup database plus archivelog;
RMAN> exit
```

MONITORING

Application server components are best monitored using Oracle Grid Control product (OEM). This is the only known tool that monitors the entire stack from top to bottom. OEM is used for monitoring as well as cloning and deployment. OEM supports functionality known as a "beacon". The "beacon" runs through a set of command and examines the result. Consider setting up

beacons to test each major component deployed. Doing so will make troubleshooting problems a simple task.

OEM also offers up tools that simplify modification of configurations. The tools from OEM and the steps to move data from one installation to another are typically sufficient for all logical use cases.

This paper does not cover the installation and setup of GRID control.

APPENDIX A - SCRIPTS

/oracle/migrate/portal/get_passwords.ksh

```
# Note, this script refrences the development environment
#    and may need changing for credentials
# Call Synytax is “get_password.ksh <USERNAME>
#
$ORACLE_HOME/bin/ldapsearch -h portal.dbaknow.com -p 389 -D cn=orcladmin -w manager1 -
b "cn=IAS,cn=Products,cn=OracleContext"  -s sub -v OrclresourceName=$1  |grep -v
orclflexattribute1 |grep -v objectclass |grep -v objectclass
```

/oracle/migrate/portal/First_Portal.ksh

```
. ./portal.env

if [ ! -d imp_log ]
then
  mkdir imp_log
fi

$MIDTIER_ORACLE_HOME/opmn/bin/opmnctl stopall

sqlplus "sys/$SYS_PASSWORD@$PORTAL_TNS as sysdba" <<IASDB

spool imp_log/drop_create_user.log

alter database flashback off;

---- Drop users
--
-- Warning: You need to stop all SQL*Plus connection to the
-- portal schema before that else the drop will give an
-- ORA-01940: cannot drop a user that is currently connected

drop user portal_public cascade;
drop user portal_app cascade;
drop user portal_demo cascade;
drop user portal cascade;

---- Recreate the users and give them grants"
--
-- The new users will have the same passwords as the users we just dropped
-- above. The schema passwords are found in OID. See: portal_automatic_env.sh
--
-- Do not forget to add your exported custom users
```

```
create user portal identified by $PORTAL_PASSWORD default tablespace portal;
grant connect,resource,dba to portal;
create user portal_app identified by $PORTAL_APP_PASSWORD default tablespace portal;
grant connect,resource to portal_app;
create user portal_demo identified by $PORTAL_DEMO_PASSWORD default tablespace portal;
grant connect,resource to portal_demo;
create user portal_public identified by $PORTAL_PUBLIC_PASSWORD default tablespace
portal;
grant connect,resource to portal_public;
alter user portal_public grant connect through portal;
start $MIDTIER_ORACLE_HOME/portal/admin/plsql/wwv/wdbigra.sql portal

exit
IASDB

exit
```

/oracle/migrate/portal/Second_DB.sh

```
. ./portal.env

imp userid="'sys/$SYS_PASSWORD@$PORTAL_TNS as sysdba'"
file=exp_data/portal_exp.dmp grants=y log=imp_log/import.log full=y

exit
```

/oracle/migrate/portal/Third_DB.sh

```
. ./portal.env

sqlplus "sys/$SYS_PASSWORD@$PORTAL_TNS as sysdba" <<IASDB

spool imp_log/sys_post_changes.log

---- Recompile the invalid packages"
--
-- On the midtier, the script utlrp is not present. This step
-- uses a copy of it stored in patch/utlrp.sql

select count(*) INVALID_OBJECT_BEFORE from all_objects where status='INVALID';
start ?/rdbms/admin/utlrp.sql
set lines 999
select count(*) INVALID_OBJECT_AFTER from all_objects where status='INVALID';

---- Jobs
--
-- Reassign the JOBS imported to PORTAL. After the import, they belong
-- incorrectly to the user SYS.
update dba_jobs set LOG_USER='PORTAL', PRIV_USER='PORTAL' where
schema_user='PORTAL';
commit;

exit
IASDB
```

/oracle/migrate/portal/Fourth_Portal.sh

```
. ./portal.env

echo "---------------------- step 6 - post import changes (PORTAL)"

sqlplus $PORTAL_USER/$PORTAL_PASSWORD@$PORTAL_TNS <<IASDB


set serveroutput on
spool imp_log/portal_post_changes.log

---- Intermedia
--
-- Recreate the portal indexes.
-- inctxgrn.sql is missing from the 9040 CD-ROMS. This is the bug 3536937.
-- Fixed in 9041. The missing script is contained in the downloadable zip file.

start $MIDTIER_ORACLE_HOME/portal/admin/plsql/wws/inctxgrn.sql
start $MIDTIER_ORACLE_HOME/portal/admin/plsql/wws/ctxcrind.sql

---- Import error
--
alter table "WWSRC_PREFERENCE$" add constraint wwsrc_preference_pk
  primary key (subscriber_id, id)
  using index wwsrc_preference_idx1
/
begin
 DBMS_RLS.ADD_POLICY ('', 'WWSRC_PREFERENCE$', 'WEBDB_VPD_POLICY',
   '', 'webdb_vpd_sec', 'select, insert, update, delete', TRUE,
   static_policy=>true);
end ;
/

---- Modify tables with full URLs
--
-- If the domain name of prod and dev are different, this step is really important.
-- It modifies the portal tables that contains reference to the hostname or port
-- of the development machine. (For more explanation: see Addional steps in the note)

-- groups (dn)
update wwsec_group$
set dn=replace( dn, '$DEV_DN', '$PROD_DN' )
/
update wwsec_group$
set dn_hash = wwsec_api_private.get_dn_hash( dn )
/
-- users (dn)
update wwsec_person$
set dn=replace( dn, '$DEV_DN', '$PROD_DN' )
/
update wwsec_person$
set dn_hash = wwsec_api_private.get_dn_hash( dn)
/
-- subscriber
update wwsub_model$
```

```
set dn=replace( dn, '$DEV_DN', '$PROD_DN' ), GUID=':1'
where dn like '%$DEV_DN%'
/
-- preferences
update wwpre_value$
set varchar2_value=replace( varchar2_value, '$DEV_DN', '$PROD_DN' )
where varchar2_value like '%$DEV_DN%'
/
update wwpre_value$
set varchar2_value=replace( varchar2_value, '$DEV_HOSTNAME:$DEV_PORT',
'$PROD_HOSTNAME:$PROD_PORT' )
where varchar2_value like '%$DEV_HOSTNAME:$DEV_PORT%'
/
-- page url items
update wwv_things
set title_link=replace( title_link, '$DEV_HOSTNAME:$DEV_PORT',
'$PROD_HOSTNAME:$PROD_PORT' )
where title_link like '%$DEV_HOSTNAME:$DEV_PORT%'
/
-- web providers
update wwpro_providers$
set http_url=replace( http_url, '$DEV_HOSTNAME:$DEV_PORT',
'$PROD_HOSTNAME:$PROD_PORT' )
where http_url like '%$DEV_HOSTNAME:$DEV_PORT%'
/
-- html links created by the RTF editor inside text items
update wwv_text
set text=replace( text, '$DEV_HOSTNAME:$DEV_PORT', '$PROD_HOSTNAME:$PROD_PORT'
)
where text like '%$DEV_HOSTNAME:$DEV_PORT%'
/
-- Portlet metadata nls: help URL
update wwpro_portlet_metadata_nls$
set help_url=replace( help_url, '$DEV_HOSTNAME:$DEV_PORT',
'$PROD_HOSTNAME:$PROD_PORT' )
where help_url like '%$DEV_HOSTNAME:$DEV_PORT%'
/
-- URL items (There is a trigger on this table building absolute_url automatically)
alter trigger wwsbr_url_trg disable
/
update wwsbr_url$
set absolute_url=replace( absolute_url, '$DEV_HOSTNAME:$DEV_PORT',
'$PROD_HOSTNAME:$PROD_PORT' )
where absolute_url like '%$DEV_HOSTNAME:$DEV_PORT%'
/
alter trigger wwsbr_url_trg enable
/
-- Things attributes
update wwv_thingattributes
set value=replace( value, '$DEV_HOSTNAME:$DEV_PORT',
'$PROD_HOSTNAME:$PROD_PORT' )
where value like '%$DEV_HOSTNAME:$DEV_PORT%'
/
commit;
exit
```

```
IASDB
```

/oracle/migrate/portal/Fifth_Portal.sh

```
. ./portal.env

# Configure portal such that portal uses the infrastructure database

cd $MIDTIER_ORACLE_HOME/portal/conf/
./ptlconfig -dad portal -pw $PORTAL_PASSWORD
cd -
mv $MIDTIER_ORACLE_HOME/portal/logs/ptlconfig.log imp_log

date
```

/oracle/migrate/portal/Sixth_DB.sh

```
# Assumes that start/stop scripts have been created and named as referenced
#
. ./portal.env

# Environment is "repdev"

/oracle/admin/ALL/stop_all.ksh
/oracle/admin/ALL/start_db.ksh

$ORACLE_HOME/bin/sqlplus "/ as sysdba" <<EOF
shutdown immediate
startup mount
alter database flashback on;
alter database open;
exit
EOF

/oracle/admin/ALL/stop_db.ksh
/oracle/admin/ALL/start_all.ksh

date
```

CONCLUSION

Oracle Fusion products can be completely redundant and resilient. Gathering the information and planning prior to installation will save many hours and many problems down the road. The breatdth and depth of Oracle's offerings is growing and it is never too late to ensure that implementations are done right.

Frank Bommarito

DBA Knowledge, Inc.
frank@dbaknow.com
http://www.dbaknow.com